IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 3, 2021, accepted June 24, 2021, date of publication June 28, 2021, date of current version July 6, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3093102

Speeding Up Bluetooth Mesh

DAVID PÉREZ-DÍAZ-DE-CERIO[1], ÁNGELA HERNÁNDEZ-SOLANA[2], MARIO GARCÍA-LOZANO[1], ANTONIO VALDOVINOS BARDAJÍ[2], AND JOSÉ-LUIS VALENZUELA[1]

[1]Signal Theory and Communications Department, Universitat Politècnica de Catalunya, 08860 Castelldefels, Spain
[2]Aragon Institute for Engineering Research (I3A), University of Zaragoza, 50018 Zaragoza, Spain

Corresponding author: David Pérez-Díaz-de-Cerio (dperez@tsc.upc.edu)

This work was supported in part by the Spanish Ministry of Science through the European Regional Development Funds (ERDF) under Project RTI2018-099880-B-C32, Project RTI2018-095684-B-I00, and Project RTI2018-099063-B-I00; and in part by the Government of Aragon (Reference Group) under Grant T31_20R.

This work did not involve human subjects or animals in its research.

ABSTRACT Bluetooth has constantly evolved from its cradle in 1997 to the last 5.2 version in 2020. With each update and amendment, it has gained in speed, range, and versatility. One of the latest introductions was the Bluetooth Mesh Profile (BMP) making it a technology suitable for a wide variety of applications. Nevertheless, BMP was designed to maintain the compatibility with Bluetooth version 4 devices already deployed in the market. This imposes some restrictions that place Bluetooth Mesh under other competing technologies like Zigbee or Thread in terms of throughput performance. In this paper we propose two mechanisms to overcome these limitations and take advantage of the new extended advertising capabilities introduced with Bluetooth 5. These mechanisms are presented as modifications to the current protocol stack to allow the transmission of larger data structures. Thus, it is possible to boost the throughput of Bluetooth Mesh making it suitable to more demanding applications like, for example, image transmission. The first proposal is designed as an adaptation layer to avoid modifying the standard in its current form. The second makes minimal changes to the frame structure at the different layers enabling the user to accommodate possible encapsulations (i.e., tunneling) without incurring IPv6-layer fragmentation. We have analyzed both solutions and compared them with the current BMP in terms of throughput, delay, and energy consumption for different channel conditions and network size. The results show that except for very small messages or poor channel conditions the proposals improve the throughput and delay of the current BMP.

INDEX TERMS Bluetooth, Internet of Things, throughput, delay, energy, wireless communication, wireless mesh networks.

I. INTRODUCTION

Wireless technologies have progressively evolved over the last 30 years. Their performance, throughput, capacity, range, reliability, and energy efficiency are improved with each new standard or revision of the existing ones. This enables the creation of new applications and services extending their use to practically any field. Wireless Sensor Networks (WSNs) are a subset of technologies initially developed for enabling data transmission using unlicensed wireless bands without a service provider. Among the usual applications for WSN, we can find many fields including smart cities, industrial, health, environmental, agriculture, transportation, military, etc [1]–[4]. In this kind of environments, it is important to consider several key points like battery life, reliability, latency, and throughput, but also if the transmission mechanism is connection-oriented or is based on broadcasting frames like flooding mechanisms [5]. Among these standards, those that use the 2.4 GHz band based on the 802.15.4 standard (Zigbee, Thread, WirelessHart) stand out over the rest [6] and now Bluetooth should also be considered in this group.

When Ericsson started the development of Bluetooth back at the beginning of the 90's, the main objective was to replace the connection cables between headphones and computers with a cellular phone. Since then, it has evolved to the current version 5.2 [7] with improved transmission speeds, energy consumption, and interoperability. One of the advantages of Bluetooth over its competitors is its widespread implantation in the market. Bluetooth is ubiquitous in cell phones,

The associate editor coordinating the review of this manuscript and approving it for publication was Ding Xu.

computers, watches, cars, health devices, sensors, etc. This is important as it enables the rapid development of new applications as demonstrated by the creation of Covid-19 case tracking applications that use Bluetooth technology [8].

Before the publication of the Mesh Profile Specification many alternatives to incorporate meshing in Bluetooth were proposed. For example, in [5] the alternatives of a connection or broadcast mode are analyzed and from all of them, an interesting proposal stands out among others: the FruityMesh algorithm [9]. This open source solution establishes a connection between the nodes which is kept open until the message reaches its destination. Another worthy reading is [10], where multiple mesh solutions are presented, some of them based on flooding mechanisms and where "the main application scenario are networks where low energy consumption is the main concern and data is non time-sensitive." Even other standardization bodies like the Internet Engineering Task Force (IETF) are trying to include mesh networking into Bluetooth, 6BLEMesh [11]. The authors that participate in the definition of this new approach have also published an experimental evaluation of this technology considering latency, roundtrip time and energy consumption in [12]. They also evaluate the impact on the performance for three different platforms (Nordic, LaunchPad and Texas Instruments) concluding that the lifetime of the devices can differ in more than 4 months depending on the manufacturer. However, these may have come to an end when, in 2017, the mesh topology was officially incorporated [13], [14] to the Bluetooth specifications allowing multi-hop communication, increasing the reach, and many-to-many communication. An interesting comparison showing the performance characteristics and the trade-offs between 6BLEMesh and the official BLE mesh can be found in [15]. Nevertheless, note that 6BLEMesh at the moment of this writing is still in a draft stage and the official BLE mesh was released in 2017 and received the last revision in 2019. Hence, due to its more mature state and official nature we used the current standard as our main reference.

The BLE mesh standard uses a managed flooding mechanism instead of routing. This mechanism is very robust, allows the immediate sending of messages, the message always arrives by the shortest path, and it does not require maintenance of the routes. This new feature has been added on top of the Bluetooth core. And, in the current version, it uses the advertising events for transmission within the mesh network as the link-layer transmission base. Specifically, the non-connectable present in version 4.0. The natural evolution of the standards makes it foreseeable that BLE mesh will incorporate improvements to the initial version.

Nevertheless, currently, technologies based on the 802.15.4 standard have better performance in terms of throughput than BLE mesh. These technologies present a maximum payload of 127 bytes [16] at MAC level, but can be used on different frequency bands: 868 MHz, 915 MHz, and 2.4 GHz. The maximum transmission at physical layer is 20 kbps, 40 kbps, and 250 kbps, respectively. However, other options may apply: the use of acknowledgments, different address fields (0, 16, 64 bits), or the use of the slotted/unslotted modes of CSMA/CA. Thus, the maximum throughput can fluctuate between 12.8 and 162 kbps [17]. Derived technologies from this standard, like Zigbee or Thread, incorporate new layers that reduce the real effective throughput and, in practice, make it implementation-dependent. For example, Burchfield *et al.* state that with an appropriate implementation one can obtain 108 kbps at the application layer from a theoretical maximum of 120 kbps using hardware that initially presented 53 kbps [18]. In fact, other experimental and more recent studies like the one made by Silicon Labs [19] show different results demonstrating that they depend on the implementation. In this case, they make a thorough comparative analysis of Zigbee, BLE, and Thread considering the influence of the data packet size and number of hops. Their results present a real measured throughput in Zigbee of 21 kbps in a single hop or 5 kbps in five hops, whereas Thread obtains near the double under the same conditions, 47 kbps and 10 kbps, respectively. All of them consider 100 bytes of user data.

In contrast, BLE mesh, which is currently based in BLE v4 advertisements, has only 29 bytes available for the payload at the advertising bearer layer, transmitted at least every 25 ms in average. If we add the necessary overhead to transmit data at the application level to the payload at the link-layer level, this difference is accentuated as the user payload increases.

Multiple papers analyze the benefits of WSN technologies. For example, the study done in [19] shows that systems based on 802.15.4 and BLE mesh have similar performance in throughput and latency for small payloads (10 bytes). However, when the payload size is increased, the Bluetooth performance is much lower than for the other two technologies considered (Thread and Zigbee). This is mainly due to the lower payload capacity of the link layer. With BLE v4 advertisements like the ones used in the current BMP, Bluetooth requires segmentation for application payloads greater than 10 bytes, while 802.15.4-based systems are around 100 bytes.

Regarding network management, the Bluetooth flooding algorithm allows easier management, greater redundancy, and faster response times concerning network or transmission failures. Additionally, it can operate without establishing a connection nor any kind of hand-shaking. However, routing-based mechanisms generate fewer packets, resulting in less interference and collision probabilities. A thorough study of the QoS performance of BLE mesh in terms of delay, scalability and reliability even in presence of outer and self-interferences [20] demonstrates that BLE mesh is particularly vulnerable to network congestion and increased packet collision probability for densely populated deployments. Nevertheless, the randomization of the timing parameters of the protocol reduces its impact.

The Bluetooth Mesh Profile is able to work on top of any version 4.0 device since its release. This allows backwards compatibility. But, as of version 5.0, the Bluetooth core adds a new advertising mode that allows the transmission of larger packets in broadcast mode, called extended advertising. Besides, this type of advertising uses the entire ISM band and allows less occupation of channels 37, 38, and 39. However, the current BMP frame structure is still based on the definitions made for v4.0. This implies that although a mesh network could be composed of newer and more powerful devices because BLE v5 devices are compatible with BLE v4 devices, it is not possible to take full advantage of their potential. So, the main research motivation for the proposals presented in this paper is to overcome this limitation that could make BLE mesh more efficient and open it to other application use cases, particularly, in planned or unplanned scenarios where devices experience good channel conditions and low interference. Under non-ideal conditions, when transmissions are affected by high bit error rates, we will see that the current BMP will be the preferred option, providing the more robust performance. Nevertheless, when better bit error rate can be achieved, its use supposes and underutilization of capacity. In this case, we show that the proposal of using the extended advertisement event provides a very significant benefit.

The industry also seems interested in taking profit of these new capabilities and, for example, Nordic Semiconductor has developed and implemented a new bearer called Instaburst [21] based on this new feature. However, this solution is proprietary and not compatible with the current standard. Furthermore, their use of extended advertising is limited to a maximum size of 498 bytes of raw advertising data.

In this paper, we propose to use extended advertising to increase the transmission of information in a Mesh BLE network. Our two proposals allow boosting the transmission speed and capacity of a Mesh network, without losing the advantages inherent to the flooding algorithm. The first is presented as an adaptation layer introduced between the network and bearer layers with the objective of not modifying anything of the current standard structure. The second proposal goes far beyond and exploits the new advertising extensions completely, but modifies the frame structure at all layers. However, these modifications affect only the PDU sizes at each layer, keeping the rest of the frame intact. Hence, although Bluetooth Mesh networks are not designed to keep a sustained high throughput and constant data transmissions, with these modifications it is possible to eventually transmit longer messages in shorter times.

Summarizing, the contributions of this paper are:

- We present two different approaches to adapt extended advertising events to the current BMP. These approaches include an Adaptation Layer (AL) that can be easily inserted in a current implementation as it does not modify the current BMP and a BLE mesh Modified Profile (MP) that improves even more the performance enabling also the transmission of large frames without the need of IPv6-layer fragmentation. Both proposals allow the use of segmentation.
- We perform a comparison of both proposals vs. the current BMP considering throughput, delay, number of hops, message size, bit error rate (BER), and energy savings.
- We determine the conditions where the new proposals, defined as an optional operation mode, offer a clearly benefit in terms of capacity enabling the support of more demanding services.
- Their application offers a delay reduction and a throughput boost in a single hop from 3.79 kbps up to 483 kbps that can be achieved for bit error rates under 10^{-7}. Additionally, transmission times can also be reduced with the corresponding energy savings that implies.

Nevertheless, the application of any of these two proposals comes with some side effects that should also be considered. Thus, to obtain favorable results all the devices in the network must support BLE version 5, channel conditions should be acceptable, and both proposals underperform for very short messages.

To address this, in the following section, it is necessary first to review the basics of Bluetooth Mesh with its main parameters, protocol stack, and frame definitions. Additionally, as our proposals are based on the new capabilities introduced with Bluetooth 5 these are also summarized and presented in section II. Thus, at this point, it is then possible to introduce our two proposals in section III. This section discusses in detail where and how to introduce both modifications into the specifications. Afterwards, in section IV, we present the comparative results obtained after applying the proposals. This section starts analyzing the throughput improvement under the best possible scenario, a single-hop and ideal channel conditions, which is the comparison reference point with other technologies. Then, we consider and discuss the effect of propagating the message through a simple linear network topology. Here, new issues appear and some parameters should be adjusted to optimize the results. The section ends with the consideration of non-ideal conditions and a detailed analysis on how several parameters impact not only on the throughput but also on other metrics like error rate, delay, and energy consumption. Finally, the paper is closed with some conclusions in section V.

II. BLUETOOTH MESH AND BLUETOOTH 5 OVERVIEW

A. BLUETOOTH MESH

A Bluetooth Mesh network enables information transmission between its nodes over a Bluetooth Low Energy physical layer. As in any mesh network, the nodes relay and cooperate dynamically to transport messages across the network.

In order to become part of the network, a device must first be provisioned by an authorized device. In this process, the device obtains the initial configuration and security credentials. Once provisioned, the device is then called a node and it is allowed to transmit and receive messages.

Additionally, a node may implement special roles that modify its behavior and operation. Hence, a node may present one or several of the following features: relay, friend, low power, and proxy. From all of them, the relay feature deserves special attention because a node with this feature enabled can retransmit messages. This allows the communication of messages beyond the first hop range and create the mesh architecture itself.

The deployment of larger networks not only implies using relays but also other procedures to ensure reliable data transmission through the network. To address this, Bluetooth Mesh uses a flooding mechanism, i.e., the messages are broadcasted and the relays retransmit the messages they receive to propagate them all over the network. However, this is done in a controlled way called *managed* flooding. The relays check two parameters before retransmitting a message: a Time To Live field (TTL) present in every message and a Network Message Cache. The TTL limits the number of times a message can be relayed and the cache prevents the relaying of previously received messages.

As the scenario can completely change from one deployment to another, the Bluetooth Mesh Profile [13] defines several parameters and procedures to control and adjust the reliability and the end-to-end delay:

- *Network Transmit:* This procedure controls the number of transmissions of data generated in a source node and the interval between them. These are controlled by the parameters network transmit count (N_{TC}) and network interval. The N_{TC} is a three-bit parameter that ranges from zero to seven, so a single message can be transmitted up to 8 times leaving a space between them equal to the network interval plus a random delay. Note that a value of $N_{TC} = 0$ means that only one message is sent. For higher values, each transmission is a verbatim copy of the initial one including the sequence number. The network interval depends on another 5-bit parameter called Network Transmit Interval Steps (N_{TIS}) that ranges from 0 to 31. The final network interval is calculated using 10 ms steps, i.e., $(N_{TIS}+1)\cdot 10$ ms. Hence, theoretically, the network interval could take values from 10 to 320 ms. However, the minimum should not be less than the minimum advertising interval which is 20 ms for the type of advertisements used in BLE mesh (see sections 7.8.5 and 7.8.53 of [7]).
- *Relay Retransmit:* It controls how many times a packet should be retransmitted by a relay and their spacing within the same range as the network transmit just presented. Its operation is very similar to the network transmit procedure but affects to the messages received via radio at the relays. The parameters involved are the Relay Retransmit Count (R_{RC}) and the Relay Retransmit Interval Steps (R_{RIS}). Again, the first one is a 3-bit parameter that indicates the number or retransmissions (up to eight) and the second (5 bits) adjusts the relay interval using also 10 ms steps with the same restrictions explained before.
- *Publish Retransmit:* It determines how many times a single message should be published (P_{RC}) and the time between these publications. This procedure introduces more redundancy publishing the same message several times ($0 \leq P_{RC} \leq 7$). Each and every publication should also follow the network transmit procedure. Thus, a message could be sent up to 64 times ($(P_{RC}+1)\cdot(N_{TC}+1)$). However, the published messages are not the same as each publication has a different sequence number. The interval between publications uses steps of 50 ms with a Publish Retransmit Interval Steps (P_{RIS}) in the range from 0 to 31 keeping the final value between 50 ms and 1600 ms.
- *Random Delay:* To avoid collisions, the mesh profile also requires introducing a random delay (T_{delay}) between transmissions. This random delay is not specified in the BMP, but a typical value derived from the core specifications for this delay is 10 ms (section 4.4.2.2.1 of [7]).

Most of these procedures are applied at the network layer (network transmit, relay transmit, random delay) or above (publish retransmit). However, this paper is focused on the layer just below it, the bearer layer.

The bearer layer defines how to transport the network messages between nodes. Currently, there are two bearers defined: the GATT (Generic Attribute) bearer and the advertising bearer.

On one hand, current devices that do not support natively the Bluetooth Mesh Profile can use the GATT bearer to communicate with a mesh network through a proxy node. And, on the other hand, when it is possible to include the mesh profile into the device stack, the advertising bearer should be used. Recall that the BMP is designed to work even on the first BLE devices, i.e., version 4.0 compliant devices.

In fact, what the specifications state is that the advertising bearer shall use non-connectable and non-scannable undirected advertising events. This leaves the door open to the use of the new extensions presented in BLE version 5.0.

Nevertheless, the frame structure defined in the upper layers follows to the letter one of the Bluetooth Mesh Profile design conditions: "it must work on existing devices in the market today" [13]. Hence, this structure is designed supposing the use of the original and most restrictive ADV_NONCONN_IND indications to keep compatibility with version 4 devices.

Figure 1 illustrates the complete encapsulation process from the application layer to the bearer layer considering ADV_NONCONN_IND based advertisements. The bottom half of the figure is dedicated to explaining the advertising bearer operation. In its final form, the information is transported within advertising events. An advertising event is, generally, composed of three ADV_NONCONN_IND each one transmitted on a different primary channel (37, 38, and 39). The information sent is the same in all three channels.

In BLE, advertising events are transmitted following a regular pattern: each one is sent after a fixed period (advertising interval) perturbed by a small random value (T_{delay}). In Bluetooth Mesh, however, this interval depends on the network transmit/relay retransmit procedures previously explained. Looking into the frame structure, ADV_NONCONN_IND indicators can transport a Bluetooth Mesh network PDU of

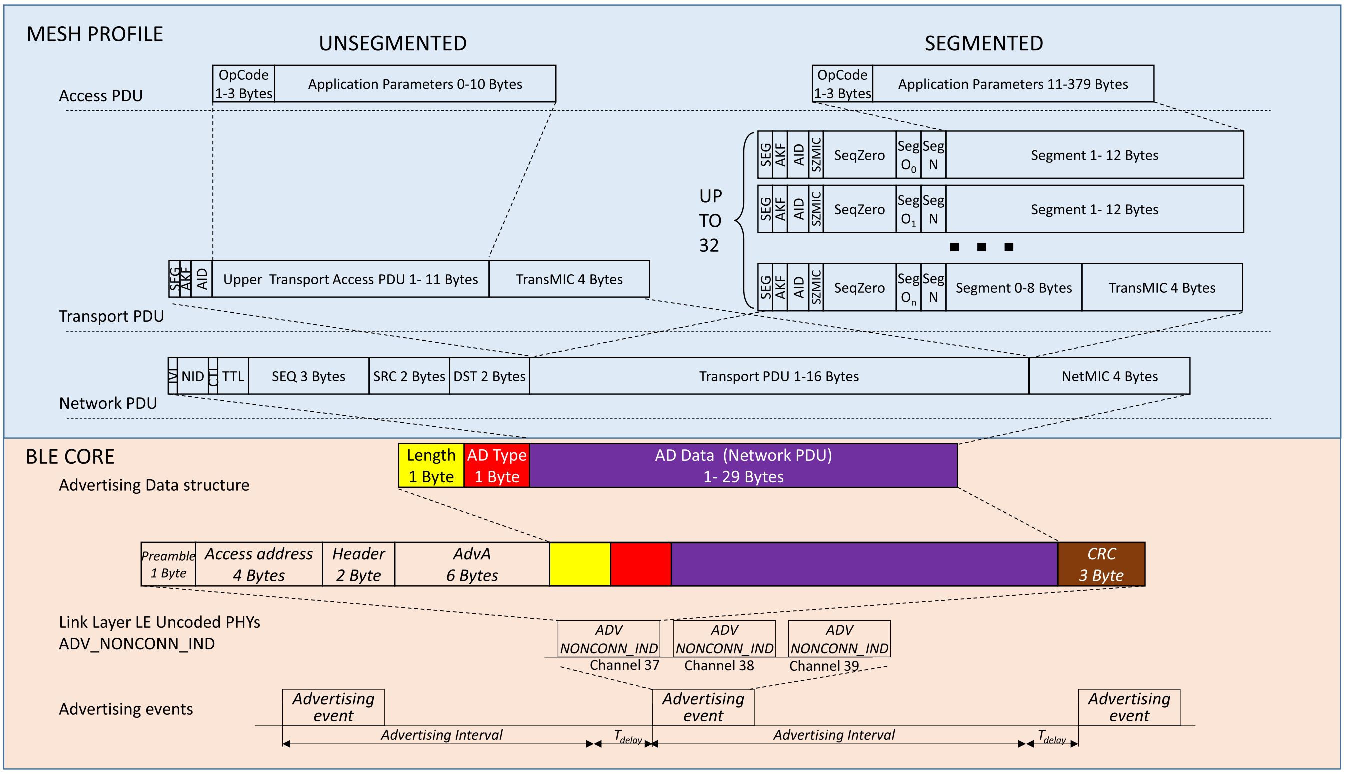


FIGURE 1. **Data encapsulation over ADV_NONCONN_IND advertisements.**

up to 29 bytes. Hence, unsegmented messages could transport up to 10 user application bytes (see Figure 1). To solve this, the Bluetooth Mesh Profile allows the transmission of longer application messages applying segmentation. In that case, up to 379 user bytes can be distributed over 32 different segments. Nevertheless, under the best conditions, each one would be transported by a single advertising event. This increases considerably the time required to transmit the data.

Thus, the decision of using ADV_NONCONN_IND implies reduced effective data throughput. Next, we will calculate the throughput for this case and we will use it as the base to compare. This effective throughput can be calculated as the relation between the total user data bits and the time needed to transmit this message before another one can be transmitted. It is important to remark the difference between this time and the time the data is effectively on the air. The latter is shorter, but the system does not allow to transmit data continuously.

For example, we will start analyzing the simplest case following the left-hand side from up to the bottom of Figure 1: 10 user data bytes without segmentation. To form the access layer PDU, the user data bytes need an operation code of at least 1 byte. This is encapsulated into the next layer PDU, the transport layer, adding its own 1-byte header and the four octets of TransMIC to secure the application data. The resulting 16 bytes pass through the network layer which adds 13 bytes more. Finally, the Advertising Data structure is built adding the length and advertising type fields before being delivered to the bearer.

At this point, the advertising bearer takes these 31 bytes and composes the ADV_NONCONN_IND, a 376-bit frame in this case. On the air, this represents 376 μs with the GFSK modulation used at 1 Mbps.

The time between the beginning of two consecutive ADV_NONCONN_IND within an advertising event shall be less than or equal to 10 ms. But in any case, the advertising event shall be closed within the advertising interval. Hence, considering the minimum advertising interval (20 ms), an average random value of 5 ms and just one transmission per packet ($P_{RC} = N_{TC} = R_{RC} = 0$), the maximum effective throughput is 3200 bps, see equation 1.

$$th_{\max} = \frac{application\ payload\ bits}{\min\left(network\ interval\right) + mean(T_{delay})} \tag{1}$$

For the segmented case, up to 379 application data bytes can be distributed over 32 segments sent every $\min(network\ interval) + mean(T_{delay})$ to obtain a maximum effective throughput of 3790 bps.

B. BLUETOOTH 5 EXTENDED ADVERTISING

Bluetooth 5 [22] introduces, among other improvements, new advertising mechanisms. To make use of the new capabilities, Bluetooth 5 defines additional Advertising Physical Channel PDUs. These allow to increase advertising PDU payloads up to 254 bytes over the secondary channels (data channels), chain several advertisements with fragmented host advertising data to transmit even more data in a single advertising event, and/or increase the transmission data rate.

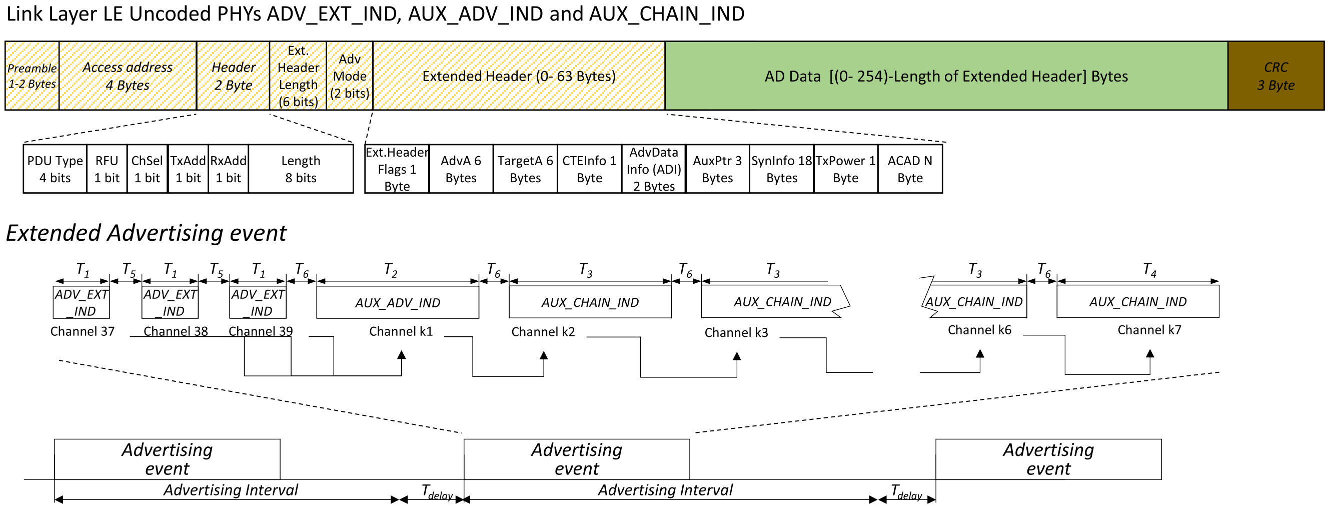


FIGURE 2. Extended advertising event and its encapsulation.

Focusing only on the ones allowed by the Bluetooth Mesh Profile, the non-connectable and non-scannable undirected advertising events, these would be the new ADV_EXT_IND, AUX_ADV_IND, and AUX_CHAIN_IND. All of them share the same payload format shown at top of Figure 2. However, the fields they use are different, and not all of them may be present. We will discuss the details later. One important example of this is the ADV_EXT_IND, which is transmitted in the primary advertising channels and is not allowed to include advertising data (AD Data). In this case, the ADV_EXT_IND would just point to an AUX_ADV_IND which would contain up to 254 bytes of AD Data.

Additionally, if the host has a large bundle of data to transmit, it can be fragmented and the AUX_ADV_IND can point to an AUX_CHAIN_IND. Then, if necessary, this AUX_CHAIN_IND could potentially link to another AUX_CHAIN_IND. This last step could be repeated until reaching the maximum of 1650 bytes of Host Data defined in [22]. This process is illustrated at bottom of Figure 2 considering that all the PDUs are filled up with the maximum AD Data bytes except the last one. In this case, this implies the transmission of one AUX_ADV_IND and six AUX_CHAIN_IND. Note also that the AUX_ADV_IND and AUX_CHAIN_IND use the secondary advertising physical channels, i.e., data channels.

Finally, another significant improvement of Bluetooth 5 is that the new extended advertisements can be transmitted at different rates: 1 Mbps, 2 Mbps, and coded for long-range. However, again, the ADV_EXT_IND is restricted and cannot be transmitted at 2 Mbps.

III. PROPOSALS

A. BLUETOOTH 5 MESH ADAPTATION LAYER (AL)

Once the new capabilities introduced with Bluetooth 5 have been detailed, we propose to take advantage and use them in Bluetooth Mesh. A first approximation could be to keep the structure of the Mesh profile without modifications and introduce a new adaptation layer between the network and bearer layers as depicted in Figure 3.

Thus, the AD Data structures would still be 31 bytes long. However, we could concatenate up to 53 AD Data structures into a single extended advertising event. Recall that this limit comes from the maximum of 1650 bytes imposed by the Host Controller Interface [18].

Hence, with 53 AD Data structures of 31 bytes, the total would be 1643 bytes. The remaining 7 bytes would not be useful because just the headers of a new message would exceed this value.

In this way, if we do not use segmentation, and remembering that in each AD data structure we can include a maximum of 10 user data bytes, up to 530 bytes of effective user data could be transmitted.

On the other hand, if segmentation is used, 379 user bytes would be distributed into 32 segments. So, from the 53 AD data structures, 21 would still be available. This results in 247 additional user bytes (20 structures of 12 bytes, plus 1 last structure of 8 bytes minus 1 byte from the OpCode). Then, in this case, up to a total of 626 effective user bytes could be sent.

A final check would be necessary as the time to transmit the 53 AD Data structures should not exceed the minimum advertising interval of 20 ms. As explained before, to transmit all the data apart from the three ADV_EXT_IND it would be required to use one AUX_ADV_IND and a maximum of six AUX_CHAIN_IND as depicted in Figure 2. All follow the frame structure of Figure 2 selecting only some of the optional fields of the extended header. We propose to keep the following:

- Extended Header Flags: required when the header is not zero.
- Advertiser Address: not present in AUX_CHAIN_IND
- Advertising Data Information (ADI): used to quickly identify a previously received message.

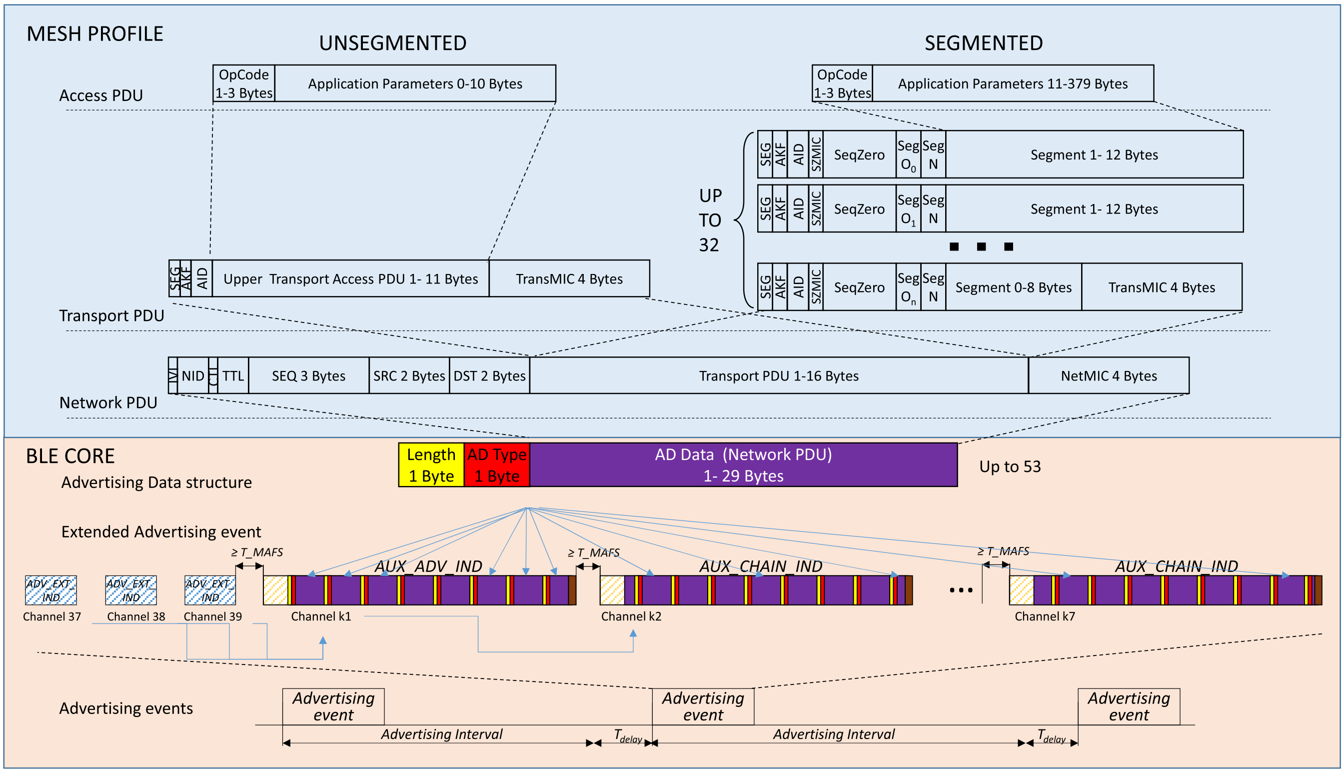


FIGURE 3. Adaptation layer for the use of extended advertising in Bluetooth Mesh.

- Auxiliary Pointer (AuxPtr): indicates the information of the next auxiliary packet. It is not present in the last AUX_CHAIN_IND.
- Transmitted Power (TxPower): although this field is optional, it is kept due to its common use in many applications.

Note that the AUX_CHAIN_IND does not contain the 6 bytes assigned to the Advertiser Address. Nevertheless, we decided that, for the sake of simplicity, all the auxiliary packets, except the last should transmit the same maximum amount of data (241 bytes, 254 - 13 header bytes). Additionally, note also that the last AUX_CHAIN_IND not only conveys less data but also lacks the AuxPtr field (3 bytes). Given all this, Table 1 summarizes the transmission times and in-between intervals for this integration of extended advertisings in the mesh stack.

According to the specifications [22], T_5 shall be greater than the Minimum Subevent Space ($T_{MSS} = 150\ \mu s$) and T_6 shall be greater than the minimum AUX Frame Space ($T_{MAFS} = 300\ \mu s$), but an exact value is not defined. To be more specific, we decided to measure these values in a real implementation using Nordic nRF52840 development kits and their Mesh SDK. The obtained values were $T_5 = 283\ \mu s$ and $T_6 = 400\ \mu s$.

The worst-case scenario would come when every frame is transmitted at 1 Mbps and, even then, the total required time would be 18174 μs. This value is below the minimum network interval of 20 ms. Hence, according to equation (1), the effective throughput for the unsegmented and segmented cases would be 169.6 kbps and 200.32 kbps, respectively.

TABLE 1. Bluetooth mesh with extended advertising timings.

	Description	Rate (Mbps)	Duration (μs)
T_1	ADV_EXT_IND duration	1	192
T_2	AUX_ADV_IND duration	1 or 2	2120 / 1064
T_3	AUX_CHAIN_IND duration	1 or 2	2072 / 1040
T_4	Last AUX_CHAIN_IND duration	1 or 2	1752 / 880
T_5	Time between ADV_EXT_IND	N/A	≥ 150
T_6	Time between auxiliary frames	N/A	≥ 300

Nordic Semiconductor is working on a similar approach called Instaburst [21]. When it is enabled, the system decides whether a message can be encapsulated in a regular advertisement or should use an extended advertisement event. However, in this latter case, Instaburst only allows one AUX_ADV_IND and one AUX_CHAIN_IND limiting the raw advertising data to 498 bytes every advertising interval. To reduce the on-air packet duration, they use an uncoded 2 Mbps bearer. Nevertheless, subtracting the overhead introduced by the upper layers of the Bluetooth Mesh stack, the effective throughput results around 60 kbps.

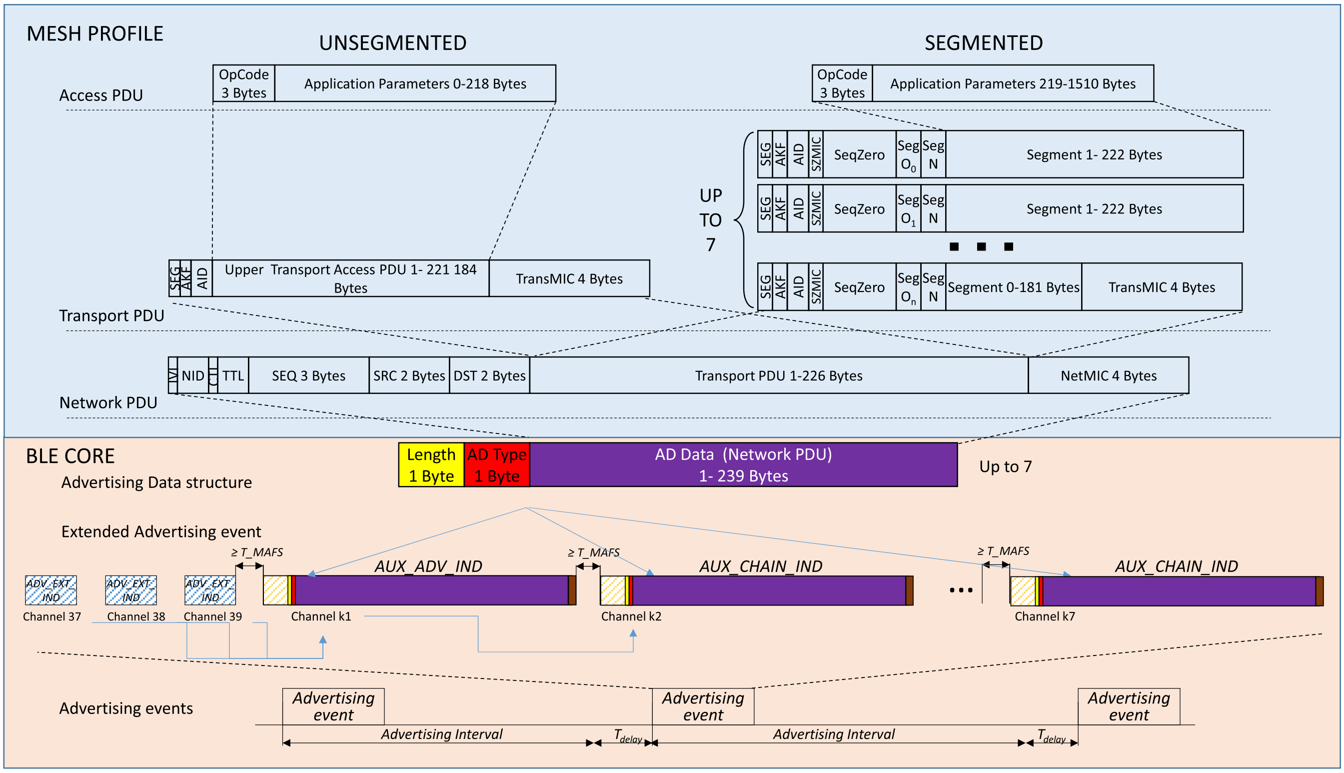


FIGURE 4. Modified mesh profile stack.

In any case, both proposals, although simple, require to update the firmware of all the devices participating in the transmissions. Thus, it would be more efficient to redesign the upper layers of the Bluetooth Mesh Profile according to the new capabilities of the extended advertisements.

B. MODIFIED MESH PROFILE FOR BLUETOOTH 5 SUPPORT (MP)

Our second proposal modifies the network, transport, and access layers to use the full capacity of the extended advertising events. This allows to reduce the overhead per packet compared with the previous proposals.

The encapsulation would be the one presented in Figure 4. Note that the frame format of the extended advertising events and their headers follow the same structure explained in section IV. In this way, the maximum network PDU inside an advertisement data structure is increased from 29 bytes to 239 bytes. At network level this implies that a frame can convey a transport PDU up to 226 bytes.

Then, if the message is unsegmented, up to 218 application bytes can be sent in a single frame without segmentation. This is considering the use of a 3-byte operation code and subtracting the 5 bytes for the transMIC and headers at the transport layer. We use the recommended manufacturer 3-byte OpCode to identify this new type of frames, but if these were introduced by the Bluetooth SIG, it could be possible to save two additional bytes in the OpCode.

On the other hand, if the message is segmented, a single segment could transport up to 222 bytes. Note that the maximum number of segments is 7 instead of 32. To explain this limit and the maximum application data size it is necessary to recall that there is a 1650 bytes maximum data size allowed by the HCI controller to form the extended advertising event. This would correspond to six full advertisement structures of 241 bytes and a last advertisement structure of 204 bytes. Hence, at transport level, the first six segments are composed of 222 bytes plus 4 header bytes and the last segment conveys a maximum of 181 bytes plus the headers and transMIC. It would be possible to reduce two bits on each of the SegO and SegN fields and keep these four bits for future use. Nonetheless, since this reduction does not reach a full byte, for the sake of simplicity, we propose to continue modifying only the PDU sizes at each layer leaving the rest of the frame intact. The 1650-bytes limit also applies to the unsegmented case. In this way, up to seven unsegmented messages could be aggregated inside a single extended advertising event having the last one a maximum of 181 application data bytes and the rest the full 218 bytes. Consequently, that would correspond to 1489 application data bytes in the unsegmented case. Nevertheless, if we allow segmentation, up to 1510 user data bytes distributed over seven segments can be inserted into a single advertising event. This is an important achievement because it overcomes the recommended value of 1500 bytes [23] and enables the user to accommodate possible encapsulations (i.e., tunneling) without incurring IPv6-layer fragmentation.

IV. RESULTS AND DISCUSSION

A. SINGLE-HOP THROUGHPUT RESULTS

With the values presented in the previous sections, and following equation (1) with a network interval of 20 ms, we will compare in this section the effective throughput for the different proposals and some state-of-the-art solutions. Initially, these results aim to determine the maximum throughput values. Thus, they are obtained considering ideal conditions with no packet losses. In a subsequent section, we will analyze the effect of introducing transmission errors.

For example, the effective throughput achieved with the modified mesh profile (section III.B) is 476 kbps for the unsegmented case and 483 kbps otherwise. This second approach is around 2.5 times faster than the best case (segmented) of our first proposal (section III.A) and more than 125 times faster than the maximum reference throughput of 3.79 kbps calculated in section II.A.

It is important to remark that such integration of extended advertisings into the Bluetooth mesh operation would position this technology much better than their current competitors: Instaburst, Thread or Zigbee. All these results are summarized in Figure 5. Please, note that the values for other technologies are extracted from [19] and [21].

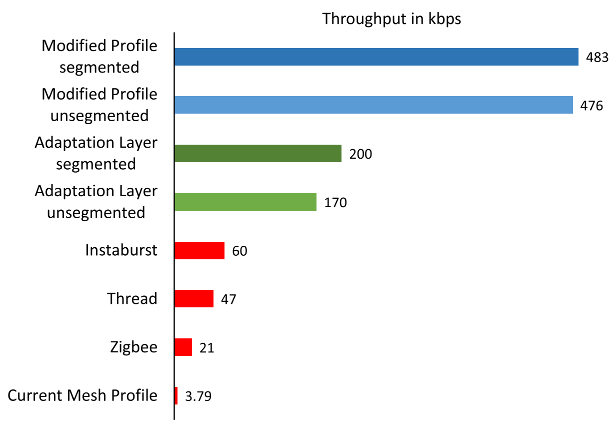


FIGURE 5. Comparison of effective throughput for a single hop.

Another and more visual way of evaluating the benefits of the proposals could be to calculate the time required to transmit a determined amount of data. To illustrate this, in Table 2 we have compiled the results with a range from a small chunk of 10-bytes data packet to a large 5 MB file in a single hop.

As can be seen, for the smallest 10-byte data chunk all the solutions perform equal. This is because all the information can be transmitted in a single advertising event in all cases.

However, with a slight increase, the standard Mesh profile quickly needs more events and, therefore, more time to transmit the same amount of data. For example, the transmission of an IPv6 packet of 1500 bytes would require 3.2 s using the current profile, but still requires a single event of 25 ms for the Modified Profile. In this case, the solution proposed

TABLE 2. Required transmission time comparison.

Message size	Mesh Profile	Instaburst	Adaptation Layer	Modified Profile
10 bytes	25 ms	25 ms	25 ms	25 ms
50 bytes	125 ms	25 ms	25 ms	25 ms
200 bytes	425 ms	25 ms	25 ms	25 ms
500 bytes	1 s	75 ms	25 ms	25 ms
1500 bytes	3.2 s	200 ms	75 ms	25 ms
100 kB	3.5 min	13 s	4 s	1.7 s
1 MB	35.2 min	2.2 min	40 s	16.5 s
5 MB	2.9 h	11.1 min	3.3 min	1.4 min

by Nordic (Instaburst) would need eight times more. Or, for instance, if we need to transmit a small 5 MB video file from a security camera, using the current Mesh profile would require nearly three hours and less than 90 seconds with our proposal.

B. DISCUSSION: MESH FLOODING AND CHALLENGING RETRANSMISSION ISSUES

No matter which proposal is selected, with the use of extended advertising a new issue arises: the transmission time is now much longer. Using regular advertising, a transmission is received within a maximum of 376 μs, whereas in the worst case, with extended advertising, it can reach near 20 ms.

This entails a potential increase in the number of collisions, error rate, packet losses, etc. But, even in an ideal scenario where the relays were aligned between source and sink without mutual interference, one should reconsider the selection of the appropriate values for the network transmit/relay retransmit intervals because a device cannot transmit and receive simultaneously.

Thus, in this and following sections, when the network presents multiple hops, the layout considered is linear and the reach of the nodes only covers their first hop. This situation is analyzed with the help of Figure 6 where the layout is depicted on the left. Let's consider a simple case with just one relay (green) between the source (yellow) and sink (black). In this example, the source would have two different messages to transmit. Each message generates a full advertising event represented as a rectangle filled with the color assigned to the transmitter and has a duration T_{event}.

For simplicity, in this section we consider error free transmissions, Bit Error Rate (BER) $= 0$. So, the network transmit count, publish retransmit count, and relay retransmit count are equal to zero. In this way, the source transmits only one event per message and the relays just perform one retransmission. The orange areas indicate the random window where a packet can start its transmission currently a value between 0 and 10 ms ($rand_{10}$). The gray rectangles depict the period during which a device is receiving a transmission.

To maximize the throughput, it is necessary to minimize the time between the transmission of two different messages

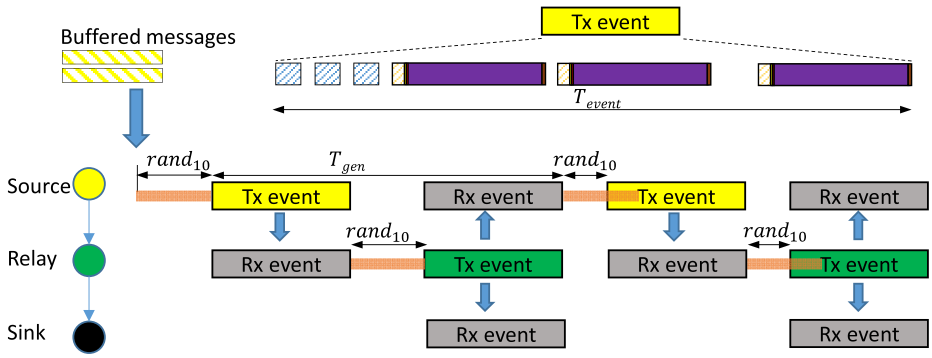


FIGURE 6. Transmission example with two hops.

at application level in the source. We call this parameter the generation interval (T_{gen}). However, note that the minimum value of T_{gen} cannot be less than the network interval and, additionally, it is also necessary to avoid simultaneous transmissions and receptions between two consecutive nodes. For example, in the case depicted in Figure 6, it has no sense to begin the transmission of the second message until the relay has finished the retransmission of the first because it could not be processed. Thus, in the worst case, T_{gen} must be at least two times the duration of the advertising event (T_{event}) plus the maximum value of the random time:

$$T_{gen} = 2 \cdot T_{event} + \max\left(rand_{10}\right) \tag{2}$$

However, this value changes if we consider more hops. For example, if we introduce two additional relays (blue and purple, in Figure 7a) the source should wait not only for the green relay to retransmit the message but also an additional time while the next relay (blue) is transmitting. At this point, the green relay would be blocked, processing the retransmission of its own message by the blue relay. Thus, it would not be prepared to receive any new incoming transmission from the source node. Note that the green relay must process that blue transmission to verify that it is not a new message that should be relayed as well. This is a mandatory action and the node must complete the reception of the message. We call this time T_{detect}: the time required to process whether the currently incoming message has previously arrived at the node.

Anyways, the source cannot send another message yet. It is also necessary to allow the propagation of the message through the network considering the random delay of the subsequent hops. To determine the worst-case, all the relays of the examples depicted in Figure 7 apply the maximum random delay for the first message and the minimum for the second. Hence, we can extend the formula of equation 2 to the more general equation 3 for two or more relays. There, N_{hops} denote the number of hops needed to reach the sink.

$$T_{gen} = 2 \cdot T_{event} + T_{detect} + \left(N_{hops} - 1\right) \cdot \max\left(rand_{10}\right) \tag{3}$$

As we have seen before, if a node must process the entire advertising event, T_{detect} could take values up to 18174 μs which corresponds with a full T_{event}, see Figure 7a. This imposes very high values on T_{gen}, which is inversely proportional to the throughput calculation, see equation (4). As a consequence, the system throughput would be impacted negatively. So, ideally, T_{gen} should be reduced as much as possible.

$$th = \frac{user_{data} \cdot 8}{\left(T_{gen} + mean\left(T_{delay}\right)\right)} \tag{4}$$

An obvious and direct way of reducing T_{event} would be to use a transmission rate of 2 Mbps when possible.

Additionally, another way of shortening T_{gen} would be to use the Advertising Data Info (ADI) field of the extended header (2 bytes) to identify earlier each transmitted message, see Figure 7b. In this way, T_{detect} could be reduced to a theoretical 890 μs or 1142 μs in practice that correspond to the duration of the three ADV_EXT_IND plus their interframe space, see Table 1.

Thus, a node only needs to decode one of the three ADV_EXT_IND indicators sent on the primary channels and check the ADI field to find out if the message has been received before. If this is the case, it would stop the current reception process and be ready for a new one. This solution is depicted in the second message transmission of Figure 7b. The impact of these changes in the throughput can be observed in Figure 8.

The analysis of Figure 8 confirms that the parameter with higher impact on the results is the number of hops between source and sink so the throughput is reduced rapidly within the first hops. Moreover, the reduction of T_{event} and T_{detect} by increasing the transmission rate and the use of the ADI field to detect previously received messages lose importance with the number of relays between source and sink. This is because T_{gen} is mainly driven by the random delay in those cases.

However, for scenarios with less than five relays (6 hops), they are quite relevant and present combined improvements in the effective throughput up to 35%. In any case, applying both changes always enhances the results and would also reduce the collision and error probability. The results are similar for both, the Adaptation Layer and the Modified Profile proposals and also follow the same trend.

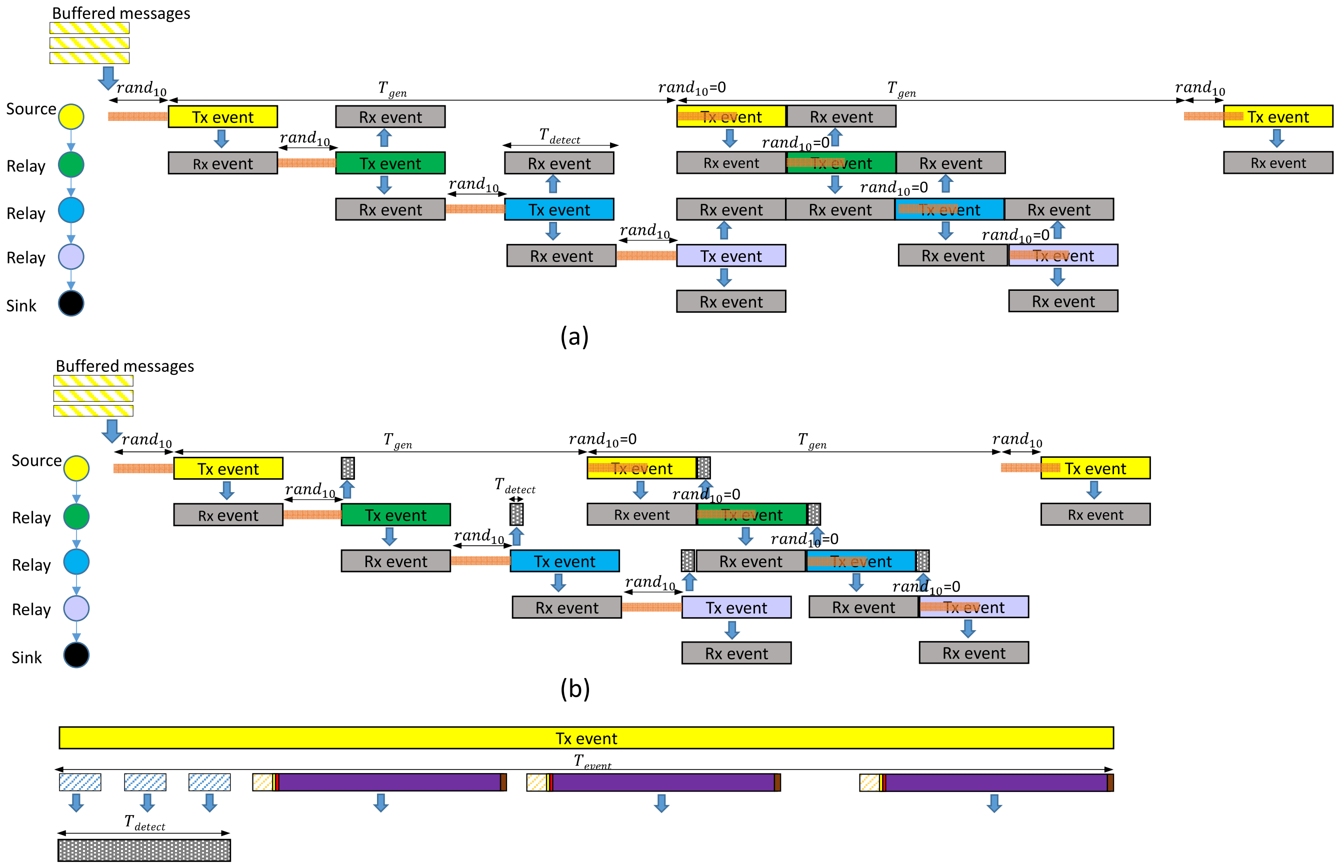


FIGURE 7. Determining T_{gen} with several hops.

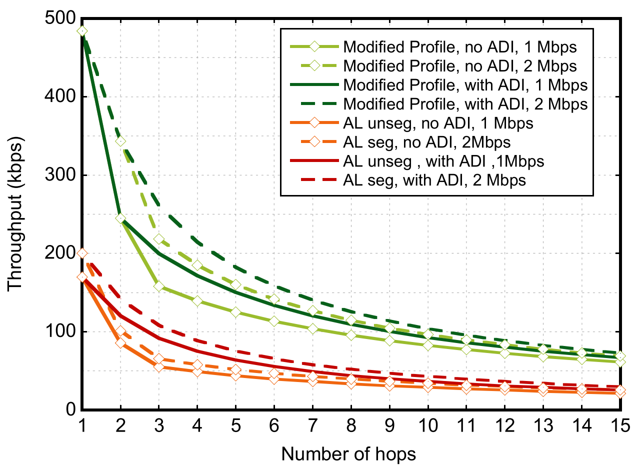


FIGURE 8. Effective throughput vs. the number of hops (ideal).

The conclusions concerning the parameters of equation (3) are also similar for both, the AL and the MP proposals and also follow the same trend as can be seen in Figure 8. However, for the AL proposal it is worth noticing the difference between using or not segmentation as depicted in Figure 8 as it becomes more relevant. The overhead reduction using segmentation improves around a 15 % the final throughput.

C. EVALUATION OF THROUGHPUT, DELAY AND ENERGY SAVINGS UNDER NON-IDEAL CONDITIONS

In order to optimally apply these proposals, the environment should present good quality links. As the messages are longer, the on-air duration is also increased. Hence, the transmissions are more vulnerable to the effect of interferences, collisions and other impairments. Thus, if any of the packets of an extended event is affected, the pointer to the next packet would be lost and also all the subsequent packets of the chain. This is aggravated with each and every hop required to reach the destination. However, in those cases, the reliability could be increased by using retransmissions, acknowledgements or limiting the number of auxiliary packets. Nevertheless, this would cause a reduction in the maximum calculated throughput as can be seen throughout this section.

To evaluate the effect of transmission impairments we consider that the received bits at each node present errors with a certain probability (BER). The origin of these errors can be diverse: fading, collisions, interferences, noise, etc. As the advertising mechanism does not provide forward error correction, a message would be wrongly received when at least one of its bits is erroneous. This is denoted as Message Error Ratio (MER). The MER is influenced by the message size, the number of hops and reduced when the messages are retransmitted.

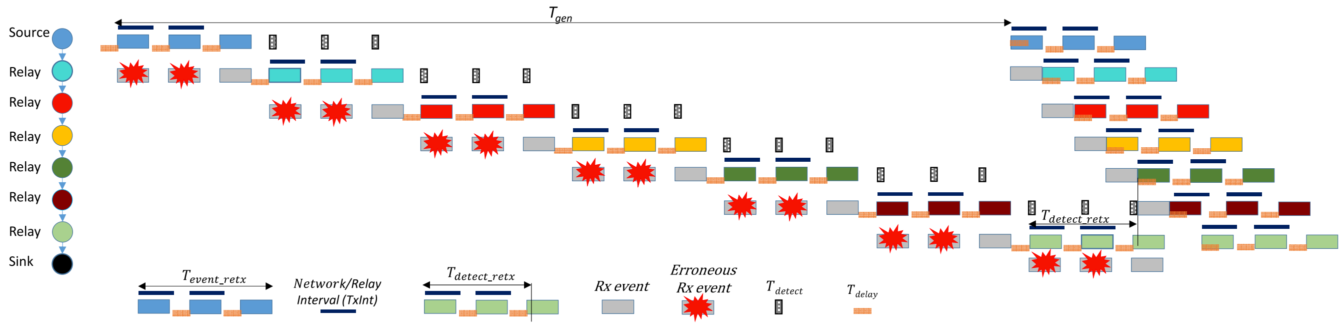


FIGURE 9. Transmission example considering non-ideal conditions and retransmissions.

In order to facilitate the evaluation and a fair comparison of the proposals, we constraint the number of variables and results by imposing a reasonable level of reception quality in terms of MER. That is, the throughput, delay and energy consumptions are calculated by imposing a 1% MER level. Thus, for a specific BER, message size and number of hops we calculate how many times a message shall be transmitted to obtain this maximum MER. The introduction of retransmissions modifies the transmission structure and the calculation method of T_{gen} and other parameters.

At the source, each message is transmitted $N_{TC} + 1$ times with a *network interval* + $rand_{10}$ period between them. At the relays the procedure is similar, but changing the parameter names. Each message is transmitted $R_{RC} + 1$ times with a *relay interval* + $rand_{10}$ period between them. Nevertheless, for the sake of simplicity, we consider $R_{RC} = N_{TC}$, so the number of transmissions (N_{TX}) at every node is $N_{TX} = N_{TC} + 1$. And, likewise, we also consider equal the *network interval* and the *relay interval*. We represent both with a new variable named *TxInt, transmission Interval*.

With the current BMP, these are all the changes to be considered, but with the new proposals T_{event} and T_{detect} become the new T_{event_retx} and T_{detect_retx}. Their definition is depicted in the legend of Figure 9 and follow equations (5) and (6).

$$T_{event\_retx} = \left(TxInt + \max(T_{delay})\right) \cdot (N_{TX} - 1) + T_{event} \quad (5)$$

$$T_{detect\_retx} = \left(TxInt + \max(T_{delay})\right) \cdot (N_{TX} - 1) + T_{detect} \quad (6)$$

Note that, if non-idealities are considered, T_{detect} should also include the time involved in frame processing at the receiver (T_{proc}) [24]. Once all these parameters are determined, it is possible to calculate their impact on the throughput, delay and energy consumption.

First, we start calculating the effect of these non-idealities on the throughput. For the current BMP, the throughput is determined by equation (7) where segmentation is also considered using the number of segments (N_{seg}) needed for transmitting a single message:

$$th = \frac{user_{data} \cdot 8}{\left(TxInt + mean\left(T_{delay}\right)\right) \cdot N_{TX} \cdot N_{seg}} \quad (7)$$

$$N_{seg} = ceil\left(\frac{user_{data} + OpCode + TransMIC}{12}\right) \quad (8)$$

For our proposals the throughput can still be calculated following equation (4). However, T_{gen} needs to be recalculated. Again, T_{gen}, depends on the number of hops and its minimum is determined for the worst-case scenario, see Figure 9.

There, considering two different consecutive messages, the first one is received always with the last retransmission after suffering the maximum random delays whereas the second is received with the first transmission and the minimum random delays at each hop. In this way, we avoid simultaneous transmissions/receptions between two adjacent nodes for any combination of random delay values and message/segment copy received. Thus, T_{gen} for a single hop follows equation (9), for two hops equation (10), and for three or more hops equation (11).

$$T_{gen} = \max\left(T_{event\_retx}, TxInt \cdot N_{TX} + \max(T_{delay})(N_{TX} - 1)\right) \quad (9)$$

$$T_{gen} = \max\left(2 \cdot T_{event\_retx} + \max(T_{delay}), TxInt\right) \quad (10)$$

$$T_{gen} = \left(T_{event\_retx} + \max\left(T_{delay}\right)\right) \cdot \left(N_{hops} - 1\right) + T_{detect\_retx} - \left(N_{hops} - 3\right) \cdot T_{event} \quad (11)$$

To determine the average delay to receive a complete message for the current BMP (d_{BMP}) we divide it into three parts. First, the delay to receive the initial segments of a message (d_{seg}). Then, the delay to receive the last segment (d_{last}). And, finally, the delay introduced at each hop of the network (d_{hop}). Note that, in average, when there are several transmissions of the same segment the received one is in the middle. Equally, from the three repetitions on the primary channels, in mean, the received one is the second. Additionally, in the calculations it is necessary to consider the interframe space (T_{IFS}) and the duration of the last transmitted PDU (T_{last}) in each channel.

$$d_{BMP} = d_{seg} + d_{last} + d_{hop} \quad (12)$$

$$d_{seg} = \left(TxInt + mean\left(T_{delay}\right)\right) \cdot N_{TX} \cdot (N_{seg} - 1) \quad (13)$$

$$d_{last} = (TxInt + mean(T_{delay})) \frac{N_{TX} - 1}{2} + 2 \cdot T_{last} + T_{IFS} + T_{proc} \quad (14)$$

$$d_{hop} = (d_{last} + mean(T_{delay}))(N_{hops} - 1) \quad (15)$$

Similarly, for the proposals presented, the average delay (d_{prop}) can be calculated using equation (16):

$$d_{prop} = T_{event} + (TxInt + mean(T_{delay})) \frac{N_{TX} - 1}{2} + \Big(mean(T_{delay}) + T_{event} + (TxInt + mean(T_{delay})) \frac{N_{TX} - 1}{2}\Big)(N_{hops} - 1) \quad (16)$$

Hence, at this point it is possible to compare the performance and delay in throughput terms against the ideal case presented in previous sections.

For example, Figure 10 depicts the combined influence of the BER and the number of hops in the maximum achieved throughput. This maximum throughput is obtained with the suitable values of retransmissions and message size and does not always correspond with the maximum message size. Note also that the maximum message size is different for each solution, ranging from the 379 bytes of the current BMP up to the 1510 bytes of the MP proposal.

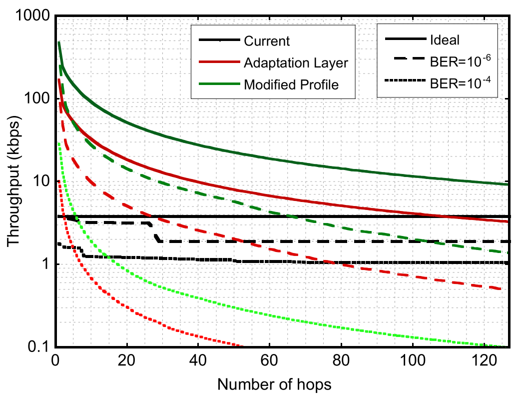


FIGURE 10. Effective throughput vs. the number of hops considering ideal and non-ideal conditions.

Observe that, for the ideal case, the MP proposal performs better than the current BMP even for the maximum number of hops allowed, 127. However, as it was expected, the benefits are reduced exponentially while the performance of the current BMP does not depend on the number of hops. On the other hand, for the AL proposal, the turning point over the current BMP appears around 107 hops. Thus, if the number of hops exceeds this threshold, this proposal should not be employed.

However, when some errors are introduced ($BER = 10^{-6}$), the performance is reduced. In this way, the MP proposal performs worse starting at 102 hops and the Adaptation Layer proposal from 52. If the conditions are worsened even more ($BER = 10^{-4}$) theses values are reduced to 15 and 4 hops, respectively. In this case, if we consider a single hop to compare the effect of these non-idealities, the throughput is 1.7 kbps for the current BMP, 10 kbps for the AL proposal and 30 kbps for the MP proposal.

Next, we are going to analyze the influence of the application PDU size under three scenarios that cover diverse possible real deployments, first for a single-hop network and then for 10-hop and 40-hop networks. Additionally, for the single hop case we are going to observe simultaneously the number of transmissions needed to keep the MER under the previously fixed 1% threshold. Notice that, at the moment a new retransmission is introduced because the 1% MER cannot be accomplished, the MER is reduced significantly and, for a while, it is possible to increase still more the message size before one more retransmission is needed. This effect can be seen in the example depicted for the MP proposal in Figure 11.

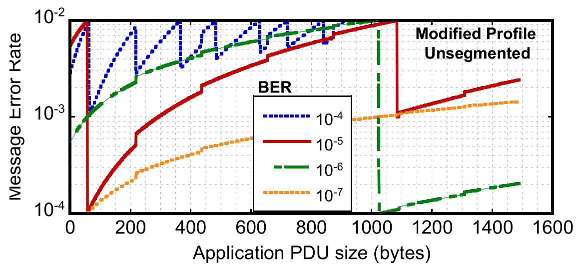


FIGURE 11. MER evolution example introducing new transmissions for a 1-hop network when the 1% limit is reached.

However, each new retransmission also augments T_{gen} which reduces the throughput and increases the delay, respectively, as could be appreciated in detail in Figures 12 and 13.

Thus, in Figure 12 we depict in pairs, one over the other, the throughput and the number of retransmissions for a determined application PDU size and a single-hop network. From upper-left to bottom-right the following results are presented: current BMP (Figure 12a), AL proposal without segmentation (Figure 12b), AL with segmentation (Figure 12c) and the MP proposal without segmentation (Figure 12d).

We intentionally left out the results for the segmented MP proposal as they are very similar to the unsegmented case.

We have checked that for BERs under 10^{-7} the simulation results can practically be considered the same as the ideal case. Hence, we take this value as the upper comparison reference.

In Figure 12a, for the current BMP, it can be observed how the throughput increases up to the 3.2 kbps calculated in section II.A for the maximum unsegmented message of 10 bytes and a BER $= 10^{-7}$(yellow line). For 11 bytes, a new segment shall be used and with it a noticeable throughput reduction. This effect can be seen with every new segment introduction, but as the new segments are filled up the throughput increases gradually up to the maximum of 3.79 kbps for a 379-byte application message. In the number of transmissions graph below this subfigure it can be observed that no retransmissions are needed to guarantee the established 1% MER.

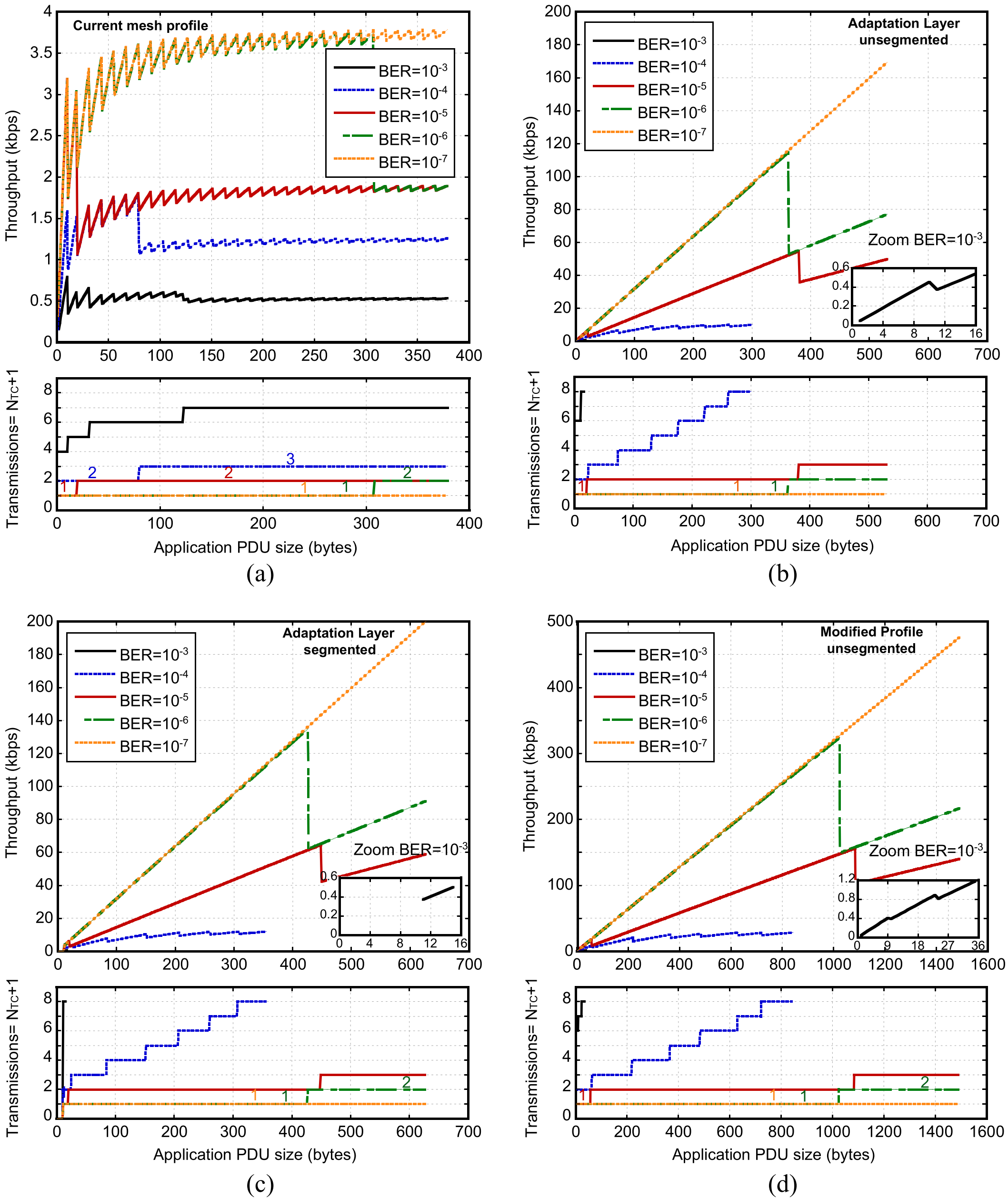


FIGURE 12. Throughput and required transmissions vs application PDU size to obtain a 1% MER under non-ideal conditions (one hop).

Thus, for worse BERs, for example 10^{-6} with the current BMP (green line in upper-left of Figure 12), it is necessary to introduce an additional transmission for application PDUs over 308 bytes and the throughput is reduced to approximately the half. Hence, T_{gen} gets increased and the maximum throughput for this scenario is achieved with 307 bytes and not for 379 bytes.

Higher BERs imply that the introduction of new transmissions is needed sooner. For example, for 10^{-5}(red), a second transmission is needed with just 18 bytes, and for 10^{-3} (black) requires four transmissions even for the shortest messages.

The rest of subfigures within Figure 12 show the results obtained for our proposals. Although all of them have a similar shape, the absolute values differ. For example, for 10^{-7}, the maximum throughput for each case is the reference value that was summarized in Figure 5: 170 kbps for the unsegmented AL, 200 kbps for the segmented AL and 476 kbps for the unsegmented MP. Observe that all these values require only one transmission for maintaining the 1% MER. However, when the BER is increased just to 10^{-6}(green) at some point, that depends on the proposal, a second transmission is needed and the throughput is drastically reduced. Hence, as we previously commented, the maximum throughput,

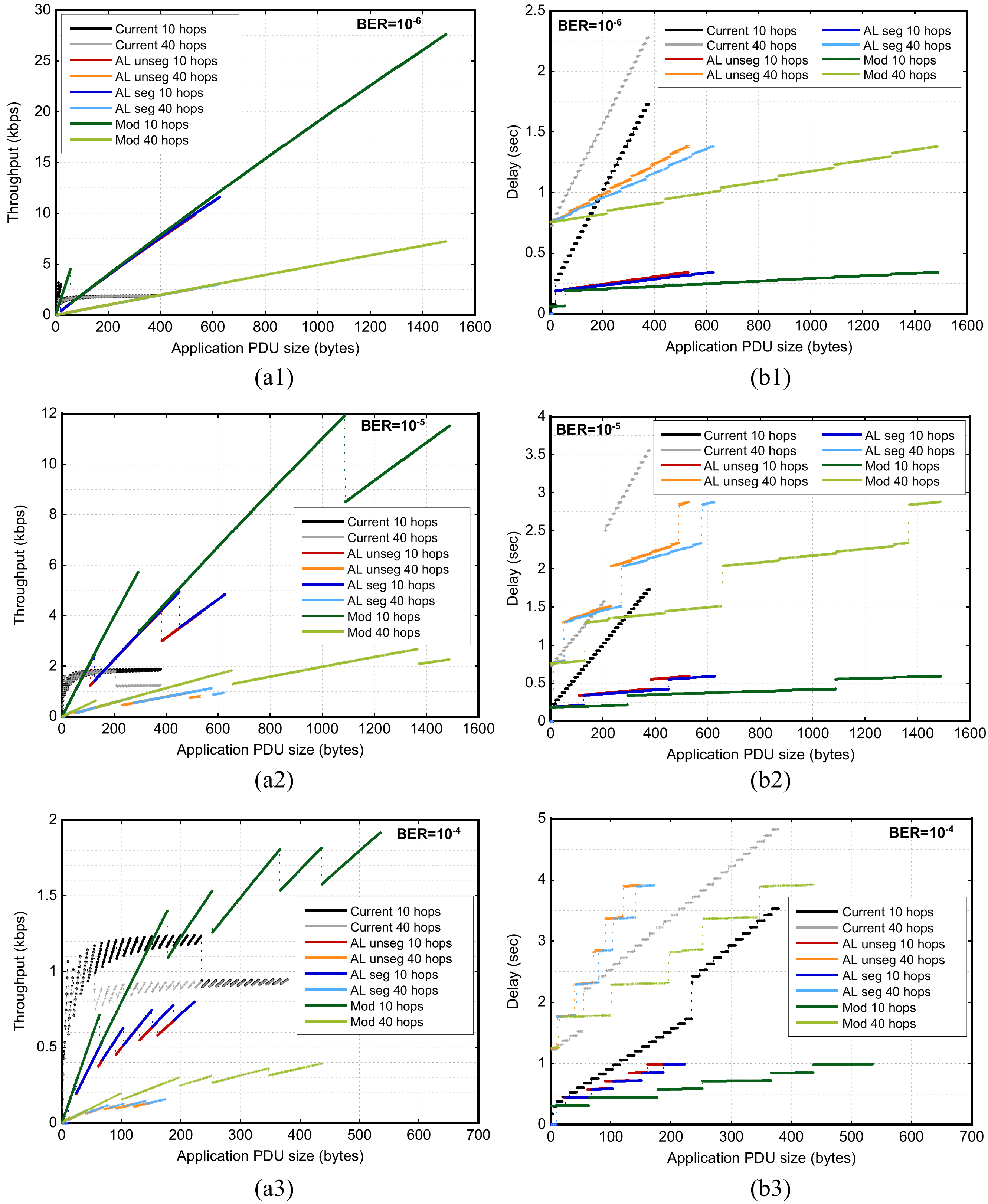


FIGURE 13. **Throughput and average delay for different channel conditions, number of hops and application PDU size.**

under this imposed constraint (1% MER) does not necessarily coincide with the maximum application PDU size.

Note also, that for higher BER values like 10^{-4} (blue), the throughput not only is considerably reduced, but also the maximum application PDU size is limited as we reach to the maximum of 8 transmissions. For example, this would be around 820 bytes for the unsegmented MP (Figure 12d) and just 300 bytes for the unsegmented AL (Figure 12b).

We also wanted to include the results for 10^{-3}, the value used at the standard to calculate the sensitivity of a Bluetooth device. These are presented in the small zoomed windows of each subfigure. As can be seen, our proposals lose most of their efficiency for this BER level and the maximum message size is very limited. For example, the AL proposal should not be applied as it clearly performs worse than the current BMP. Nevertheless, the advantages of the new proposals are significative even for small application PDU sizes (around 15 bytes) when the channel conditions improve. Hence, in a carefully planned deployment with low interference, commuting to an operation mode that uses these alternatives would provide a clear benefit.

On the other hand, Figure 13 compares the influence of the hops number and BER over the throughput and average delay for the current BMP, AL proposal, both segmented and unsegmented and for the MP unsegmented proposal.

Regarding the throughput (left column), as commented before, when the BER increases the throughput diminishes drastically, especially for our proposals. This effect is even more noticeable with the increment of hops. Thus, for 10 hops, when the channel presents errors, but the conditions are still good ($BER = 10^{-6}$) our proposals obtain much better results than the current BMP, except for small application PDUs. However, for 40 hops, the current BMP performs better in all its application range (size $<$ 380 bytes) and the benefits of our proposals only appear with longer sizes. The application range of the other proposals depends on the previously fixed 1% MER. That is the reason why, for example, the Adaptation Layer segmented (blue line) does not present values for data sizes longer than 650 bytes. This is especially remarkable in the last subfigure ($BER = 10^{-4}$) where the errors make impossible to transmit messages longer than 540 bytes no matter the proposal employed. With this BER it is recommended to use the current BMP in most of the cases and only the Modified Profile proposal can be useful in some of them when the number of hops is small. In fact, for 40 hops the throughput of both proposals grows very slowly and requires longer messages to become efficient.

In conclusion, the presented proposals provide significant enhancements when the interference conditions are acceptable. In general, this means that the BER should be under 10^{-5}. On the other hand, when the number of hops is low the improvement is remarkable even for moderated message sizes. However, if the number of hops is increased, our proposals should be selected only when the messages to be transmitted are large.

The second column of Figure 13 allows us to analyze the impact of the same parameters on the delay. Recall that we define the delay as the average time between the transmission of the first fragment of a message and the successful reception

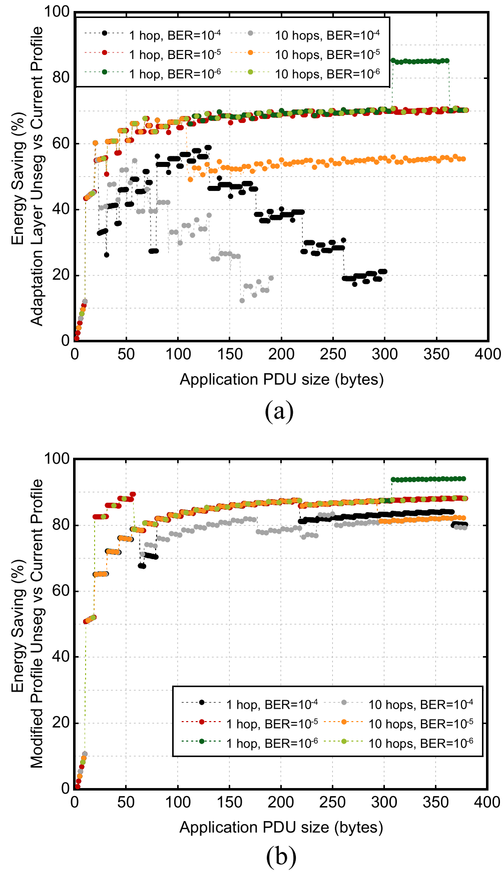


FIGURE 14. **Transmission energy savings keeping a 1% MER.**

of the last segment that completes the message as for equations (12) and (16).

It can be observed that it increases linearly with the application PDU data size presenting small steps with each new segment introduced and more noticeable jumps with each retransmission needed. Although is hardly to appreciate in the figures, again, with very small data messages (under 4 bytes), the current BMP performs better than our proposals. In this case, the delay is lower as the current BMP transmits the information directly in the primary channels while our proposals need to transmit at least one PDU in a secondary channel. However, for the rest of the cases our proposals introduce less delay, specially in good channel conditions ($BER = 10^{-6}$).

For worse BERs (10^{-4}), illustrated in Figure 13-b3, the results confronting the current BMP and the proposals in the 10-hops case get closer. However, for 40 hops, while the Modified Profile still introduces less delay than the current BMP for large messages, the Adaptation Layer proposal always performs worse.

The subfigures for $BER = 10^{-5}$ in both columns show an intermediate point and allow us to visualize the evolution from one situation to the other.

The last results, presented in Figure 14, show the energy savings between the current BMP vs. the two proposals. Figure 14a, corresponds to the AL proposal and Figure 14b, to the MP proposal. Both, consider different BER levels and depict the results for one and ten hops. The comparison is done in percentage versus the current BMP. Thus, a 60% in the figure means that the proposal analyzed needs a 60% less energy than the current BMP. We calculate the energy necessary for transmitting a message with a 1% MER for every solution. Realize that, the energy consumption has been calculated considering only the consumption related to the transmission of the messages and not the scanning periods. Hence, the energy savings are proportional to the reduction of the transmission duration.

Thus, with the current BMP, a simple message is sent using three PDUs in the three primary channels, but this needs to be multiplied by the number of transmissions required to guarantee the 1% MER.

So, the PDU size ranges from 304 μs (1-byte message) to 376 μs (10-bytes message). On the other hand, the two proposals send three AUX_EXT_IND in the primary channels (192 μs each) plus the corresponding AUX_ADV_IND and AUX_CHAIN_IND. Hence, in the minimum case, only one AUX_ADV_IND of 344 μs is transmitted. Instead, for the maximum case, one AUX_ADV_IND and six AUX_ADV_CHAIN with the values presented in Table 1 would be needed. It is necessary to remark that the presented results are calculated for a 1 Mbps transmission rate. For 2 Mbps, the transmission times would be shorter and, therefore, the energy savings would be slightly improved, but also the bit error rate would be worse.

As can be observed, the proposals always present savings over the current BMP that vary depending on the message size, number of hops and BER. For example, as depicted in Figure 14b, for a 300-bytes message the MP presents energy savings around the 80%. Only for 1, 2, or 3-byte application PDUs the current BMP performs better although it cannot be appreciated in the figure.

Again, for worse BERs or higher number of hops the benefits are reduced. And, also, it can be noticed the effect of introducing new retransmissions and/or segments.

V. CONCLUSION

The Bluetooth Mesh Profile opens the possibility to apply Bluetooth in areas and applications never seen before.

However, one of its design requirements was to be compatible with even the first versions of Bluetooth Low Energy. This prevents from taking advantage of the new capabilities introduced with the last versions of the standard and leaves Bluetooth Mesh below other similar technologies like Zigbee or Thread.

As the Bluetooth Mesh Profile should also evolve as the Bluetooth core does, in this paper we have proposed two ways of modifying it to enhance its performance. Both solutions are based on the use of the new extended advertising modes presented with Bluetooth 5.

The first one tries to avoid modifying the current profile and introduces an adaptation layer between the network and bearer layer. This layer can aggregate up to 53 advertising messages in a single extended advertising event reducing the time required to transmit the same amount of information.

The second one suggests how to make minimal changes to the base stack of the mesh profile having in mind that the bearer will use extended advertising. This is done allowing longer PDUs in every layer of the profile.

The performance of both proposals has been analyzed in terms of throughput, delay, and energy savings under ideal and non-ideal conditions. The results show that with these modifications the effective throughput in a single-hop under ideal conditions can be increased from 3.79 kbps to near 500 kbps. This demonstrates that Bluetooth Mesh could outperform other similar solutions. Nevertheless, the benefits are reduced when we consider more realistic channel conditions. In fact, it is not recommended to consider these proposals in scenarios with high BER values (BER $= 10^{-3}$) where the current BMP is more robust, but as an alternative to enhance capacity when the BER can be reduced. Both proposals require a BER under 10^{-4} or even 10^{-5} if the network presents several hops. Logically, they are also not suitable for very short messages (under around 15 bytes), but they offer considerable benefits for longer messages. The longer, the better.

Another important added value is that the Modified Profile proposal allows transmitting transparently 1500-bytes IPv6 frames. This enables Bluetooth Mesh as a suitable technology for the Internet of Things, Industrial IoT and wireless sensor networks.

REFERENCES

[1] L. Chettri and R. Bera, "A comprehensive survey on Internet of Things (IoT) toward 5G wireless systems," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 16–32, Jan. 2020.

[2] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 2015.

[3] G. A. Akpakwu, B. J. Silva, G. P. Hancke, and A. M. Abu-Mahfouz, "A survey on 5G networks for the Internet of Things: Communication technologies and challenges," *IEEE Access*, vol. 6, pp. 3619–3647, 2018.

[4] J. Weiss. (2015). Wireless sensor networking for the industrial IoT. Linear Technology. Accessed: Jun. 18, 2020. [Online]. Available: https://www.linear.com/smartmesh.

[5] R. T. E. Dvinge, A. Stalmach, and L. Nalpantidis, "Connection-based Bluetooth mesh network as a low energy solution for off-grid data networks," in *Proc. 8th Int. Conf. Modern Circuits Syst. Technol. (MOCAST)*, May 2019, pp. 1–6.

[6] D. Raposo, A. Rodrigues, S. Sinche, J. S. Silva, and F. Boavida, "Industrial IoT monitoring: Technologies and architecture proposal," *Sensors*, vol. 18, no. 10, p. 3568, Oct. 2018.

[7] Bluetooth SIG. (2019). *Bluetooth Core Specification 5.2*. Accessed: May 17, 2021. [Online]. Available: https://www.bluetooth.com/specifications/specs/core-specification/

[8] N. Ahmed, R. A. Michelin, W. Xue, S. Ruj, R. Malaney, S. S. Kanhere, A. Seneviratne, W. Hu, H. Janicke, and S. K. Jha, "A survey of COVID-19 contact tracing apps," *IEEE Access*, vol. 8, pp. 134577–134601, 2020.

[9] M-Way Solutions GmbH. (2019). *FruityMesh—The First Completely Connection-Based Open Source Mesh on Top of Bluetooth Low Energy (4.1/5.0 or Higher)*. Accessed: Jun. 18, 2020. [Online]. Available: https://github.com/mwaylabs/fruitymesh

[10] S. M. Darroudi and C. Gomez, "Bluetooth low energy mesh networks: A survey," *Sensors*, vol. 17, no. 7, p. 1467, 2017.

[11] C. Gomez, S. Darroudi, T. Svolainen, and M. Spoerk. (2021). *IPv6 Mesh Over Bluetooth(R) Low Energy Using IPSP*. Accessed: May 13, 2021. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-6lo-blemesh-10

[12] S. M. Darroudi and C. Gomez, "Experimental evaluation of 6BLEMesh: Ipv6-based BLE mesh networks," *Sensors*, vol. 20, no. 16, pp. 1–24, Aug. 2020.

[13] *Mesh Profile Bluetooth Specification V1.0.1*, Bluetooth SIG, Kirkland, WA, USA, 2019.

[14] *Mesh Model Bluetooth Specification V1.0.1*, Bluetooth SIG, Kirkland, WA, USA, 2019.

[15] S. M. Darroudi, C. Gomez, and J. Crowcroft, "Bluetooth low energy mesh networks: A standards perspective," *IEEE Commun. Mag.*, vol. 58, no. 4, pp. 95–101, Apr. 2020.

[16] *IEEE Standard for Low-Rate Wireless Networks*, IEEE Standard 802.15.4-2015. Accessed: May 17, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/7460875

[17] C. Fragouli, D. Katabi, A. Markopoulou, M. Medard, and H. Rahul, "Wireless network coding: Opportunities & challenges," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Oct. 2007, pp. 1–8.

[18] R. Burchfield, S. Venkatesan, and D. Weiner, "Maximizing throughput in ZigBee wireless networks through analysis, simulations and implementations," in *Proc. Int. Workshop Localized Algorithms Protocols (WSNs)*, 2007, pp. 15–29.

[19] *AN1142: Mesh Network Performance Comparison*, Silicon Labs, Austin, TX, USA, 2020.

[20] R. Rondon, A. Mahmood, S. Grimaldi, and M. Gidlund, "Understanding the performance of Bluetooth mesh: Reliability, delay, and scalability analysis," *IEEE Internet Things J.*, vol. 7, no. 3, pp. 2089–2101, Mar. 2020.

[21] Nordic Semiconductor. (2020). *Instaburst: Nordic Semiconductor Infocenter*. Accessed: May 17, 2021. [Online]. Available: https://infocenter.nordicsemi.com/index.jsp?topic=%2Fcom.nordic.infocenter.meshsdk.v5.0.0%2Fgroup__INSTABURST.html

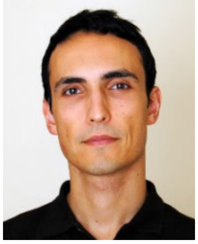
[22] Bluetooth SIG. (2016). *Bluetooth Core Specification 5.0*. Accessed: Aug. 15, 2017. [Online]. Available: https://www.bluetooth.com/specifications/adopted-specifications

[23] S. Deering and R. Hinden, *Internet Protocol, Version 6 (IPv6) Specification*, document RFC 8200, 2017.

[24] D. P.-D. D. Cerio, Á. Hernández, J. Valenzuela, and A. Valdovinos, "Analytical and experimental performance evaluation of BLE neighbor discovery process including non-idealities of real chipsets," *Sensors*, vol. 17, no. 3, p. 499, Mar. 2017.

DAVID PÉREZ-DÍAZ-DE-CERIO received the M.Sc. degree in telecommunications engineering, in 2003, and the Ph.D. degree in telecommunication from the Technical University of Catalonia (UPC), in 2010. In 2003, he joined the WiComTec Research Group, Signal Theory and Communications Department, as a Collaborating Lecturer. All his lectures are held at the Escola d'Enginyeria de Telecomunicació i Aeroespacial de Catalunya for bachelor's, second-cycle, and MAST master's students. He has participated as a consultant in several local and European projects, in addition to other public and private funded projects. His research interests include wireless communications systems, especially those based on the IEEE 802.X standards and their use in e-health applications, and the IoT.

ÁNGELA HERNÁNDEZ-SOLANA received the degree in telecommunications engineering and the Ph.D. degree from the Universitat Politècnica de Catalunya (UPC), Spain, in 1997 and 2005, respectively. She has been working at UPC and UNIZAR, where she has been an Associate Professor, since 2010. She is a member of the Aragón Institute of Engineering Research (I3A). Her research interests include 5G/4G technologies, heterogeneous communication networks, and mission-critical communication networks, with emphasis on transmission techniques, radio resource management and quality of service, mobility management, and planning and dimensioning of mobile networks.



MARIO GARCÍA-LOZANO is currently pursuing the Ph.D. degree in telecommunications engineering with the Universitat Politècnica de Catalunya (UPC, Barcelona-TECH). He has more than 20 years of experience in different radio network planning and optimization activities both at the academia and industry. He has actively participated in more than 25 competitive research projects and contracts with the industry. He is currently an Associate Professor at UPC. His research interests include radio resource management and the optimization wireless networks. He was a recipient of three best paper awards and was an Advisor of the Student Team that won the international competition for mobile network planning organized by the company ATD.

ANTONIO VALDOVINOS BARDAJÍ received the degree in telecommunications engineering and the Ph.D. degree from the Universitat Politècnica de Catalunya (UPC), Spain, in 1990 and 1994, respectively. He has been working with UPC and the University of Zaragoza (UZ), where he has been a Full Professor, since 2003. His research interests include 5G/4G technologies, heterogeneous communication networks, and mission-critical communication networks, with emphasis on transmission techniques, radio resource management and quality of service, mobility management, and planning and dimensioning of mobile networks.

JOSÉ-LUIS VALENZUELA received the degree in telecommunications engineering and the Ph.D. degree from the Universitat Politecnica de Catalunya (UPC), Barcelona, Spain, in 1993 and 1997, respectively. He is currently an Associate Professor with the Signal Theory and Communications Department, UPC. After graduation, he was concerned with equalization techniques for digital systems. He has also been working on the field of digital communications with particular emphasis on digital radio and its performance under multipath propagation conditions. His research interests include wireless sensor networks and wireless communications systems.